

SSCM Exercise 8

Nikolaus Czernin - 11721138

```
library("tidyverse")
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.5
## v forcats    1.0.0      v stringr   1.5.1
## v ggplot2    3.5.1      v tibble    3.2.1
## v lubridate  1.9.3      v tidyr     1.3.1
## v purrr      1.0.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(knitr)
```

```
# Custom print function
print_ <- function(...) print(paste(...))

set.seed(11721138)
```

```
library(ISLR)
library(Pareto)
library(HDInterval)
library(rstanarm)
```

```
## Loading required package: Rcpp
```

```
## This is rstanarm version 2.32.1
```

```
## - See https://mc-stan.org/rstanarm/articles/priors for changes to default priors!
```

```
## - Default priors may change, so it's safest to specify priors, even if equivalent to the defaults.
```

```
## - For execution on a local, multicore CPU with excess RAM we recommend calling
```

```
##   options(mc.cores = parallel::detectCores())
```

```
library(bayess)
```

```
## Loading required package: mnormt
```

```

## Loading required package: gplots

##
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
##
##      lowess

## Loading required package: combinat

##
## Attaching package: 'combinat'

## The following object is masked from 'package:utils':
##
##      combn

```

```
library(MCMCpack)
```

```

## Loading required package: coda

## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##      select

## ##
## ## Markov Chain Monte Carlo Package (MCMCpack)

## ## Copyright (C) 2003-2025 Andrew D. Martin, Kevin M. Quinn, and Jong Hee Park

## ##
## ## Support provided by the U.S. National Science Foundation

## ## (Grants SES-0350646 and SES-0350613)
## ##

##
## Attaching package: 'MCMCpack'

## The following object is masked from 'package:bayess':
##
##      rdirichlet

```

```
library(INLA)
```

```
## Loading required package: Matrix
```

```
##
```

```
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
```

```
##
```

```
##      expand, pack, unpack
```

```
## Loading required package: sp
```

```
## This is INLA_24.06.27 built 2024-06-27 02:47:30 UTC.
```

```
## - See www.r-inla.org/contact-us for how to get help.
```

```
## - List available models/likelihoods/etc with inla.list.models()
```

```
## - Use inla.doc(<NAME>) to access documentation
```

```
library(MASS)
```

```
library(bayesreg)
```

```
## Loading required package: pgdraw
```

```
## Loading required package: doParallel
```

```
## Loading required package: foreach
```

```
##
```

```
## Attaching package: 'foreach'
```

```
## The following objects are masked from 'package:purrr':
```

```
##
```

```
##      accumulate, when
```

```
## Loading required package: iterators
```

```
## Loading required package: parallel
```

```
library(boot)
```

```
##
```

```
## Attaching package: 'boot'
```

```
## The following object is masked from 'package:rstanarm':
```

```
##
```

```
##      logit
```

```
library(MCMCvis)
library(dplyr)
library(invgamma)
```

```
##
## Attaching package: 'invgamma'

## The following objects are masked from 'package:MCMCpack':
##
##      dinvgamma, rinvgamma
```

Task 1

```
# number of observations and positives for Austria
n1 <- 1279
p1 <- 0
# number of observations and positives for Germany
n2 <- 4068
p2 <- 4
reweight <- 1/10
```

Here I prepare the outlining numbers of the problem.

Building a prior distribution out of the Germany data

4 positive cases out of 4068, reweighted by a factor of 1/10

```
a_prior <- p2 * reweight + 1
b_prior <- (n2 - p2) * reweight + 1

xrate <- seq(0, .01, length.out=300)
density_prior <- dbeta(xrate, shape1=a_prior, shape2=b_prior)
```

To build the prior distribution, I take the number of successes and failures, which correspond to the alpha and beta, and multiply them with the given factor of 0.1. With those two parameters, I get a density of 300 values. I add a value of 1, which regularizes the special case of either parameter being equal to zero.

Building & plotting the posterior distribution of the Austrian data

To infer the posterior distribution of the lacking Austria data, I add the number of successes and failures to the parameters of the prior. In this case, the number of successes is 0 and the number of failures is the full observation count of the Austria data. I then use those parameters, again, to get a density over the same value range.

```
a_posterior <- a_prior + p1
b_posterior <- b_prior + (n1 - p1)
density_posterior <- dbeta(xrate, shape1=a_posterior, shape2=b_posterior)
```

```

# plot both densities
plot(xrate, density_posterior,
     main = "Density of Prior vs Posterior Distribution",
     xlab = "Number of positive cases / number of observations",
     ylab = "density",
     type = "l", col = "blue", lwd = 2)
lines(xrate, density_prior, col = "red", lwd = 1)
legend("topright", legend = c("posterior", "prior"), col = c("blue", "red"), lwd = 1)
# get the high density intervals at the 95% confidence intervals
hpdi <- HPDinterval(as.mcmc(qbeta(c(0.025, 0.975), a_posterior, b_posterior)))
print_("Bounds of the 95% density interval:", hpdi[1] %>% round(4), "---", hpdi[2] %>% round(4))

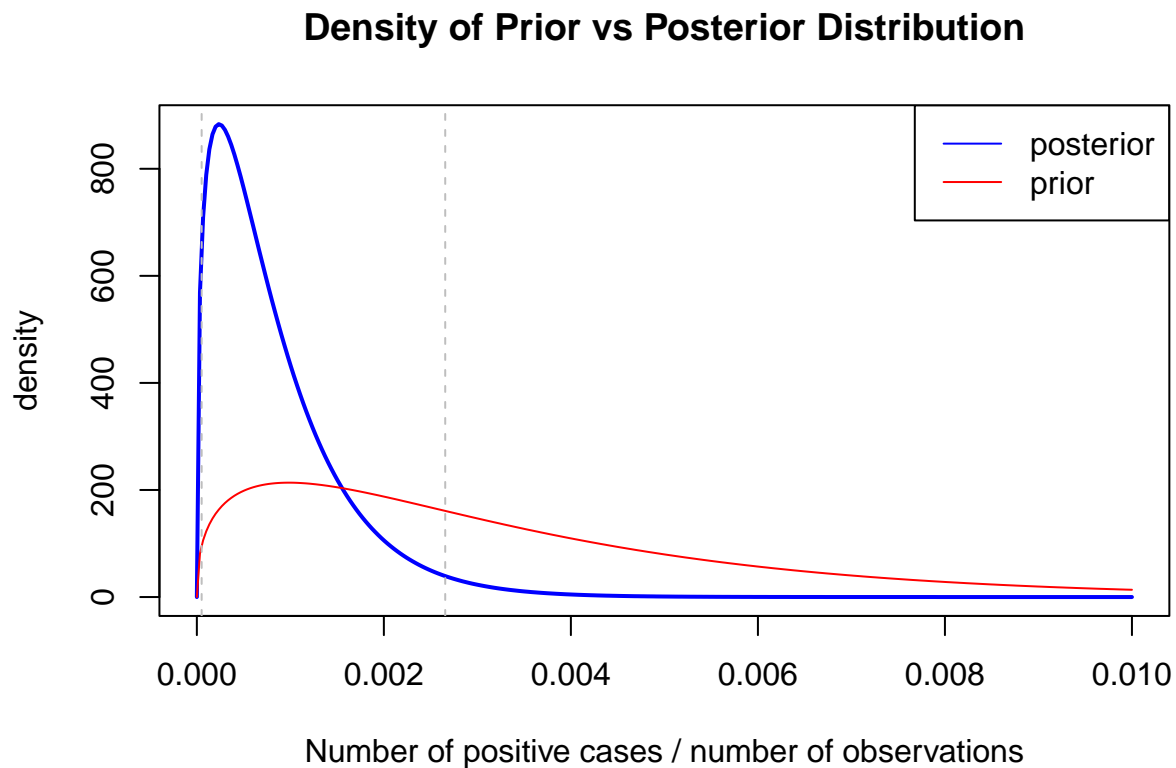
```

```
## [1] "Bounds of the 95% density interval: 1e-04 --- 0.0027"
```

```

# plot the outlines of the interval
abline(v=hpdi[1], col="gray", lty=2)
abline(v=hpdi[2], col="gray", lty=2)

```



Task 2

Since the residuals of a model are assumed to be normally distributed, we assume the data to have this normal distribution:

$$y \sim N(x^T \beta, \sigma^2)$$

the first parameter of which is the response of the linear model with the coefficients β and the second of which is the variance of the residuals. This is the likelihood function.

When we assume β as a normally distributed prior,

$$\beta \sim N(\mu_\beta, \sigma_\beta^2)$$

we can conjugate a normally distributed posterior of

$$\beta|y \sim N(\mu_{posterior}, \sigma_{posterior}^2)$$

To create a prior for the variance of the model, we have to pick a distribution that has only positive values, because variance is always positive, like for example the Gamma distribution.

Define conjugate priors

```
# beta

data(Auto)

x <- Auto$horsepower # input data
y <- Auto$mpg # the response

# scale the data
x <- x %>% scale()
y <- y %>% scale()

n <- y %>% length()
```